

IN THE CLAIMS:

The following claim listing replaces all previous claim listings.

Claim 1. (currently amended) A method for dynamically adjusting a workload of an active ~~resource~~ server, ~~the workload being expressed as a collection of units, each unit including its own key identifier, the active resource being associated with at least one parent workload group, the parent workload group including a collection of workload units such that workload units belonging to the parent workload group share an identical sequence of values at a specified depth value of their key identifiers, the identical sequence of values defining a group key identifier associated with the parent workload group,~~ the method comprising:

dividing the workload into a collection of workload units, each unit including its own key identifier;

associating the active server with at least one parent workload group, the parent workload group including the collection of workload units such that the collection of workload units belonging to the parent workload group share an identical sequence of values at a specified depth value of their key identifiers, the identical sequence of values defining a group key identifier associated with the parent workload group;

independently determining by the active ~~resource~~ server that an overload condition exists at the active ~~resource~~ server;
and

if the overload condition exists:

increasing the depth value of the parent workload group

such that at least two child workload groups are identified; and assigning a target ~~resource~~ server to manage at least one of the child workload groups.

Claim 2. (currently amended) The method of claim 1, further comprising if the overload condition exists, identifying at least one candidate ~~resource~~ server to which the child workload groups may be distributed using a decentralized protocol, the at least one candidate server including the target server.

Claim 3. (currently amended) The method of claim 1, further comprising requesting workload acceptance from the target ~~resource~~ server at a peer level.

Claim 4. (currently amended) The method of claim 1, further comprising recording the parent workload group as inactive at the active ~~resource~~ server.

Claim 5. (previously presented) The method of claim 1, further comprising transferring application-specific objects corresponding to the child workload groups at a peer level.

Claim 6. (currently amended) The method of claim 1, further comprising redirecting entities operating on elements of the parent workload group to the target ~~resource~~ server managing the child workload group.

Claim 7. (currently amended) The method of claim 6, further comprising:

receiving a probe message from an entity operating on ~~a~~ an entity workload unit, the entity workload unit being ~~that is~~ a member of the parent workload group, the probe message including a ~~guessed~~ selected key identifier ~~key~~ formed by ~~guessing~~ selecting a depth to be associated with the entity workload unit's key identifier; and

sending a response to the entity indicating the group key identifier that ~~the a~~ current ~~resource~~ server locally determines to be the nearest known active parent group to which the element's key identifier belongs.

Claim 8. (currently amended) The method of claim 7, wherein the entity operating on a workload unit uses the response to further refine its estimate of a correct depth to be associated with the unit's key identifier; and

probing another ~~resource~~ server associated with the parent key group formed by using the refined depth of the unit's key identifier.

Claim 9. (currently amended) The method of claim 1, further comprising:

determining that an under-load condition exists at the active ~~resource~~ server; and

if the under-load condition exists:

identifying at least two workload groups for consolidation

into a consolidated workload group;

generating a consolidated key identifier such that workload units belonging to the consolidated workload group share an identical sequence of values at a specified depth value of the consolidated key identifier; and

managing the consolidated workload group by the active ~~resource~~ server.

Claim 10. (original) The method of claim 9, wherein generating the consolidated key identifier includes decreasing the depth value of the parent workload group such that the consolidated workload group is identified.

Claim 11. (original) The method of claim 1, further comprising associating the workload unit with the key identifier such that the key identifier encodes one or more attributes of the workload unit.

Claim 12. (currently amended) The method of claim 1, further comprising constructing a virtual key for mapping to the target ~~resource~~ server, wherein the virtual key includes a number of masked of digits, the number of masked digits lead-dependent subset of a complete key identifier dependent on the overload condition.

Claim 13. (currently amended) The method of claim 12, further comprising using the constructed load-dependent virtual

key as an input to a separate mapping service that returns the identity of the target ~~resource~~ server to which the workload elements belonging to the virtual key should be directed.

Claim 14. (currently amended) A system for running a distributed computer application whose workload can be decomposed into a set of workload units, each workload unit including its own key identifier, over a dynamically varying set of distributed resources, the number of distributed resources involved in the distributed computer application ~~computation~~ varying dynamically in response to changes in an overall workload, the system comprising:

a set of active resources cooperatively managing an entire set of identifier keys ~~key-identifiers~~ constituting the overall workload, each individual active resource managing a dynamically varying group of identifier keys ~~key-identifiers~~, each active resource independently evaluating its own workload condition and deciding on the creation or consolidation of ~~group-key identifiers~~ identifier keys to reduce or increase its workload;

an overall set of resources, of which the active resources constitute a subset that can be utilized as part of the distributed computer application as needed;

a mapping service configured to receive a virtual key associated with at least one of the dynamically varying group of identifier keys ~~key-identifier~~ as input and configured to produce an ~~the~~ identity of a ~~the~~ target resource from the overall resource set as an output; and

a set of client entities utilizing the distributed computer application, each client entity being associated with at least

one identifier key ~~identifier~~, and each client entity dynamically determining the dynamically varying group of key identifiers ~~load-dependent group of identifier keys~~ that it ~~each such~~ currently belongs to.

Claim 15. (currently amended) A system for dynamically adjusting a workload of an active resource, the system comprising:

a plurality of workload units, each workload unit being associated with a identifier key ~~identifier~~;

a parent workload group including ~~a collection~~ the plurality of workload units, wherein the workload units belonging to the parent workload group share an identical sequence of values at a specified depth value of the identifier key ~~identifier~~; and

an active resource managing the parent workload, the active resource configured to increase the depth value of the parent workload group such that at least two child workload groups are identified and assign a target resource to manage at least one of the child workload groups if an overload condition exists at the active resource.

Claim 16. (currently amended) The system of claim 15, wherein the active resource is further configured to identify at least one candidate resource to which the child workload groups may be distributed using a decentralized protocol, the at least one candidate resource including the target server.

Claim 17. (previously presented) The system of claim 15, wherein the active resource is further configured to request workload acceptance from the target resource at a peer level.

Claim 18. (previously presented) The system of claim 15, wherein the active resource is further configured to record the parent workload group as inactive at the active resource.

Claim 19. (previously presented) The system of claim 15, wherein the active resource is further configured to transfer application-specific objects corresponding to the child workload groups at a peer level.

Claim 20. (original) The system of claim 15, wherein the active resource is further configured to identify at least two workload groups for consolidation into a consolidated workload group, generate a consolidated key identifier such that workload units belonging to the consolidated workload group share an identical sequence of values at a specified depth value of the consolidated key identifier, and manage the consolidated workload group by the active resource if an under-load condition exists.

Claim 21. (original) The system of claim 20, wherein the active resource is further configured to decrease the depth value of the parent workload group key identifier such that the consolidated workload group is identified.

Claim 22. (original) The system of claim 15, wherein the active resource is further configured to associate the workload unit with the key identifier such that the key identifier encodes one or more attributes of the workload unit.

Claim 23. (currently amended) The system of claim 15, wherein the active resource is further configured to construct a virtual key for mapping to the target resource, wherein the virtual key includes a number of masked of digits, the number of masked digits dependent on the overload condition ~~load-dependent subset of a complete key identifier.~~

Claim 24. (original) The system of claim 15, further comprising an external service configured to identify at least one candidate resource to which the child workload groups may be distributed.

Claim 25. (currently amended) A computer program product embodied in a tangible media comprising:

computer readable program codes coupled to the tangible media for dynamically adjusting a workload of an active resource, ~~the active resource being associated with at least one parent workload group, the parent workload group including a collection of workload units, each workload unit being associated with a key identifier such that workload units belonging to the parent workload group share an identical~~

~~sequence of values at a specified depth value of the key identifier, the identical sequence of values defining a group key identifier associated with the parent workload group, the computer readable program codes configured to cause the program to:~~

divide the workload into a collection of workload units, each unit including its own key identifier;

associate the active resource with at least one parent workload group, the parent workload group including the collection of workload units such that the collection of workload units belonging to the parent workload group share an identical sequence of values at a specified depth value of their key identifiers, the identical sequence of values defining a group key identifier associated with the parent workload group;

determine that an overload condition exists at the active resource; and

if the overload condition exists:

increase the depth value of the parent workload group such that at least two child workload groups are identified; and

assign a target resource to manage at least one of the child workload groups.

Claim 26. (currently amended) The computer program product of claim 25, further comprising program code configured to identify at least one candidate resource to which the child workload groups may be distributed if the overload condition exists using a decentralized protocol, the at least one candidate resource including the target server.

Claim 27. (previously presented) The computer program product of claim 25, further comprising program code configured to request workload acceptance from the target resource at a peer level.

Claim 28. (original) The computer program product of claim 25, further comprising program code configured to record the parent workload group as inactive at the active resource.

Claim 29. (previously presented) The computer program product of claim 25, further comprising program code configured to transfer application-specific objects corresponding to the child workload groups at a peer level.

Claim 30. (original) The computer program product of claim 25, further comprising program code configured to redirect entities operating on elements of the parent workload group to the target resource managing the child workload group.

Claim 31. (currently amended) The computer program product of claim 30, further comprising program code configured to:
receive a probe message from an entity operating on elements of the parent workload group, the probe message including a ~~guessed~~ selected key identifier ~~key~~ at a ~~guessed~~ selected depth; and

send a response to the entity indicating the target resource managing the child workload group.

Claim 32. (original) The computer program product of claim 25, further comprising program code configured to:

determine that an under-load condition exists at the active resource; and

if the under-load condition exists:

identify at least two workload groups for consolidation into a consolidated workload group;

generate a consolidated key identifier such that workload units belonging to the consolidated workload group share an identical sequence of values at a specified depth value of the consolidated key identifier; and

manage the consolidated workload group by the active resource.

Claim 33. (original) The computer program product of claim 31, wherein the program code to generate the consolidated key includes program code to decrease the depth value of the parent workload group such that the consolidated workload group is identified.

Claim 34. (original) The computer program product of claim 25, further comprising program code configured to associate the workload unit with the key identifier such that the key identifier encodes one or more attributes of the workload unit.

Claim 35. (original) The computer program product of claim

25, further comprising program code configured to construct a virtual key for mapping to the target resource, wherein the virtual key includes a hash value of the key identifier.